

# Clase 1 — 17.10.25

---

 Desarrollo de interfaces. JAVASCRIPT, JQUERY, REALIDAD VIRTUAL

 Clase 1 — 17/10/2025

 Tema: Presentación de la Asignatura y Introducción al concepto de interfaz

---

## Presentación de la Asignatura

**Fecha:** 17/10/2025

**Profesora:** Sara Gonzalo

**Asignatura:** Desarrollo de interfaces (Continuación de *Lenguaje de Marcas*)

---

### 1 Introducción general

La asignatura se centra en el **manejo del DOM, AJAX y repositorios de datos XML/JSON**, además de una introducción a **PHP y MySQL**.

El objetivo principal es desarrollar aplicaciones web dinámicas, accediendo y manipulando los elementos HTML desde **JavaScript**, y conectando con fuentes de datos externas.

---

### 2 Acceso al DOM desde JavaScript

- El **DOM (Document Object Model)** representa todos los elementos del documento HTML como objetos.
  - Desde **JavaScript** se puede acceder, modificar y crear esos elementos.
  - En fases avanzadas, será posible **construir una página completa únicamente desde JS**, sin escribir directamente el HTML.
  - El HTML servirá solo para **vincular la hoja de estilos (CSS) y el script principal (JS)**.
  - Toda la estructura y contenido se generarán dinámicamente mediante código.
- 

### 3 AJAX — Conexiones asíncronas con el servidor

- **AJAX (Asynchronous JavaScript and XML)** permite realizar **comunicaciones con el servidor sin recargar la página**.
  - Se basa en abrir un **hilo asíncrono** que mantiene la conexión activa mientras se reciben o envían datos.
  - Mejora la **velocidad y la experiencia de usuario**, ya que evita actualizaciones constantes de toda la página.
- 

### 4 Repositorios de datos: XML y JSON

#### ◆ Concepto general

- Se trabajará con **repositorios de datos externos**, no con bases de datos tradicionales.

- Los datos estarán almacenados en ficheros **XML** y **JSON**, por ejemplo: `alumnos.xml` o `alumnos.json`.

## ◆ Funcionamiento

- Desde **JavaScript** se crearán los **nodos** del documento y se construirá la página en función de ellos.
- Se establecerá una **conexión asíncrona con el servidor** (mediante AJAX) para leer los archivos.

## ◆ Ventajas e inconvenientes

Ventajas	Inconvenientes
Mayor rapidez al no abrir una base de datos.	Método menos seguro (sin servidor, logs ni permisos de acceso).

 **Nota:** La parte de **JSON** es la más utilizada y actual en el desarrollo web moderno.

## 5 Introducción a PHP y MySQL

- Con **PHP** se establecerán conexiones con bases de datos **MySQL**.
- El usuario podrá **loguearse** en el sistema y los datos se mostrarán dinámicamente en la interfaz web.
- Esta parte permitirá combinar la **interactividad de JS** con la **persistencia de datos del servidor**.

## 6 Interfaces, UX y Accesibilidad

### ◆ Concepto de interfaz

La **interfaz** es la capa visual e interactiva que conecta al **usuario** con la **aplicación o sistema**.

Define **cómo el usuario navega, introduce datos y recibe información**.

Incluye todos los elementos gráficos y de interacción: menús, botones, formularios, iconos, colores, tipografía y organización del contenido.

Su objetivo principal es **facilitar la comprensión, reducir errores y mejorar la eficiencia del uso**.

En desarrollo web, la interfaz se diseña combinando **HTML (estructura)**, **CSS (estilo)** y **JavaScript (interactividad)**.

### ◆ Tipos de interfaz

Tipo	Descripción	Ejemplo
<b>Gráfica (GUI)</b>	Basada en elementos visuales e iconos.	Navegadores, sistemas operativos, aplicaciones web.
<b>De línea de comandos (CLI)</b>	Basada en texto y comandos escritos por el usuario.	Terminal de Linux, CMD, PowerShell.
<b>Táctil o gestual</b>	Interacción mediante gestos o pantallas táctiles.	Móviles, tablets, cajeros.
<b>De voz (VUI)</b>	Control mediante lenguaje natural o comandos de voz.	Asistentes virtuales como Alexa o Siri.

### ◆ UX (User Experience)

**El diseño de experiencia de usuario (UX)** busca crear productos **útiles, accesibles y agradables**.

Abarca desde la **estructura de navegación** hasta la **respuesta emocional** del usuario frente a la interfaz.

Principios básicos de UX:

1. **Claridad:** el usuario debe entender rápidamente cómo usar la aplicación.
2. **Consistencia:** mantener patrones visuales y de comportamiento uniformes.
3. **Retroalimentación:** informar al usuario de lo que ocurre (mensajes, animaciones, cambios de estado).
4. **Eficiencia:** minimizar pasos y evitar acciones redundantes.
5. **Control:** el usuario debe poder deshacer errores o volver atrás fácilmente.

💡 En la asignatura, la UX se aplicará en la planificación de la **hoja de ruta de navegación**, es decir, los caminos que sigue el usuario dentro de la aplicación.

## ◆ Accesibilidad web (WAI/WCAG – W3C)

La **accesibilidad** garantiza que **todas las personas**, incluidas aquellas con discapacidades visuales, auditivas o motoras, puedan usar la web.

### Organismos y pautas principales

- **W3C (World Wide Web Consortium):** organismo internacional que define los estándares web.
- **WAI (Web Accessibility Initiative):** grupo del W3C que desarrolla las pautas de accesibilidad (WCAG).
- **WCAG (Web Content Accessibility Guidelines):** especifica los criterios que debe cumplir un sitio accesible.

### Niveles de conformidad

Nivel	Descripción	Ejemplo
A	Nivel básico de accesibilidad.	Texto alternativo en imágenes.
AA	Nivel intermedio (recomendado).	Contraste de color adecuado, navegación por teclado.
AAA	Nivel avanzado.	Subtítulos, transcripciones, navegación totalmente inclusiva.

### Principios WCAG

1. **Perceptible:** el contenido debe poder ser percibido por todos.
2. **Operable:** la interfaz debe poderse manejar con diferentes dispositivos o métodos.
3. **Comprensible:** la información debe ser clara y predecible.
4. **Robusto:** compatible con tecnologías de asistencia (lectores de pantalla, ampliadores, etc.).

⚙️ En la práctica, se evaluará el cumplimiento de estos principios y niveles mediante herramientas de validación del **W3C** y revisiones manuales de accesibilidad y usabilidad.

## 7 Proyecto de asignatura

- A partir de **febrero de 2026** se iniciará el **proyecto práctico**.
- Se establecerá una **hoja de ruta con entregas parciales** para su desarrollo progresivo.

## 8 Información administrativa

- **Duración:** clases planificadas hasta **enero de 2026**.
- **Cambios o avisos:** se comunicarán por la **mensajería interna de la plataforma**.
- **Consultas rápidas:** enviar correo a  
 sgonzalo@digitechfp.com
- En el **Foro** se publicarán anuncios generales.

Perfecto.

Aquí tienes el **punto 9** completo — “**Introducción al concepto de interfaz**”, con **toda la teoría ampliada** y organizada según las diapositivas y tus notas de clase:

## Introducción al concepto de interfaz

### ◆ Contexto y orientación de la asignatura

La profesora subraya la **importancia del proyecto final**, que se iniciará en **febrero**, aunque el **examen tendrá mayor peso en la nota final**.

El proyecto no debe tomarse como algo secundario, ya que **simula un entorno laboral real**: desarrollo, estructura, pruebas y presentación profesional.

Por ello, el temario teórico se completará antes de febrero para poder dedicar las últimas semanas a la fase práctica.

### ◆ Evolución de las interfaces

Las interfaces han evolucionado desde los primeros **sistemas operativos basados en texto** hasta los actuales entornos gráficos e interactivos.

Este avance está directamente relacionado con las **capacidades del hardware**: cada mejora visual o funcional implica **mayor consumo de recursos del sistema**.

La evolución de las interfaces busca **simplificar la interacción** sin perder funcionalidad, equilibrando **rendimiento y experiencia de usuario**.

### ◆ Ejemplos de análisis visual

#### 1. Tetera y cafetera:

- La cafetera comunica visualmente su función; la tetera del ejemplo no.
- Una interfaz debe ser **autoexplicativa**, es decir, el usuario debe entender su uso sin leer instrucciones.

#### 2. Señales confusas:

- Una mala organización visual o exceso de información **enturbia la comprensión**.
- En interfaces digitales ocurre lo mismo: demasiados elementos provocan **ruido visual**.

#### 3. Secadores de manos:

- El **tipo de usuario** y el **contexto** determinan las prioridades del diseño (rapidez, higiene, facilidad de uso).
- Los modelos más exitosos son **intuitivos y prácticos**, sin necesidad de instrucciones.

#### 4. Tarjetas de embarque:

- Ambas contienen la misma información, pero en la tarjeta mejor diseñada se **resaltan los datos relevantes**: número de vuelo, puerta, asiento y hora.
- El diseño no debe ser decorativo, sino **funcional y jerárquico**.

#### 5. Programas con exceso de iconos:

- Un diseño saturado confunde y frena la productividad.
- La **simplicidad visual** mejora la claridad y la eficacia del uso.

### ◆ Concepto de interfaz

Una **interfaz** es el espacio donde se producen las **interacciones entre el usuario y un sistema** (software o hardware).

Su objetivo es **facilitar la comunicación**, permitiendo que el usuario controle y reciba información del sistema de manera clara y eficiente.

Componentes principales:

- **Elementos visuales**: botones, iconos, menús, formularios, ventanas.
- **Elementos de control**: teclado, ratón, pantallas táctiles, gestos.
- **Feedback**: respuestas del sistema (mensajes, animaciones, sonidos, cambios visuales).

### ◆ Tipos de interfaz de usuario

Tipo	Descripción	Ejemplo
GUI (Graphical User Interface)	Interfaz gráfica con elementos visuales.	Escritorio de Windows, Android, macOS.
CLI (Command Line Interface)	Interacción mediante comandos de texto.	Terminal de Linux, CMD, PowerShell.
VUI (Voice User Interface)	Interacción por voz mediante comandos hablados.	Siri, Alexa, Google Assistant.
NUI (Natural User Interface)	Uso de gestos y movimientos naturales.	Consolas con sensores, pantallas táctiles.

### ◆ Experiencia de Usuario (UX)

La **UX (User Experience)** se refiere a la **percepción y respuesta emocional** que un usuario tiene al interactuar con un sistema o producto.

Incluye la utilidad, facilidad de uso, accesibilidad, atractivo visual y la satisfacción global que genera el sistema.

### Componentes de la experiencia de usuario

Componente	Descripción
Utilidad	Evalúa si el producto satisface una necesidad y cumple su propósito.
Usabilidad	Mide la facilidad con la que los usuarios pueden aprender y usar el producto.
Deseabilidad	Abarca los aspectos emocionales y estéticos del diseño.
Accesibilidad	Garantiza el acceso de usuarios con diferentes capacidades.
Credibilidad	Evalúa la confianza en el producto o empresa.

Componente	Descripción
Valor	Determina si el producto cumple con las expectativas del usuario.

## Factores que influyen en la UX

- **Individuales:** emociones, expectativas, experiencias pasadas, motivación.
- **Culturales:** normas, símbolos, lenguaje o religión.
- **Contextuales:** entorno, tiempo disponible, temperatura o acompañamiento.

## ◆ Diseño Centrado en el Usuario (UCD)

El **UCD (User-Centered Design)** es un proceso de diseño que coloca al usuario en el centro de todas las decisiones.

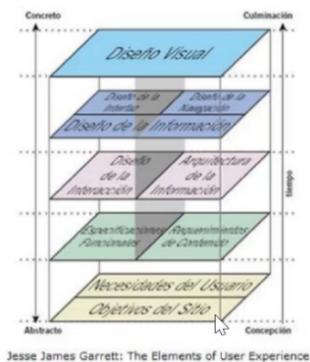
Su finalidad es garantizar que el producto final **responda a las necesidades reales** del usuario y mejore su experiencia global.

## Principios del UCD

1. **Enfoque en el usuario y la tarea:** comprensión profunda de quién usa el sistema y para qué.
2. **Participación activa:** los usuarios colaboran durante el diseño mediante entrevistas o pruebas.
3. **Iteración del diseño:** se crean y refinan prototipos en ciclos sucesivos.
4. **Diseño holístico:** combina estética, funcionalidad, accesibilidad y contexto.

## ◆ Etapas del Diseño Centrado en el Usuario

(Según Jesse James Garrett – *The Elements of User Experience*)



### 1. Investigación del usuario:

- Recolección de información mediante encuestas, entrevistas y estudios de campo.
- Creación de *personas* y escenarios de uso.

### 2. Definición de requisitos:

- Establecer los requisitos funcionales y no funcionales.
- Utilizar *mapas de empatía* para identificar prioridades.

### 3. Diseño y prototipado:

- Creación de **wireframes** (esquemas visuales de la interfaz).
- Desarrollo de prototipos interactivos para pruebas tempranas.

### 4. Pruebas de usabilidad:

- Evaluar cómo interactúan los usuarios con el prototipo.
- Aplicar mejoras según los resultados obtenidos.

## 5. Implementación y desarrollo:

- Colaborar con los desarrolladores para mantener la coherencia entre diseño y código.
- Realizar pruebas continuas durante el desarrollo.

## 6. Lanzamiento y monitoreo:

- Desplegar el producto final y recopilar *feedback* post-lanzamiento para futuras iteraciones.

 Garrett representa estas fases desde lo **abstracto (necesidades del usuario)** hasta lo **concreto (diseño visual final)**, en una estructura jerárquica que culmina en el **diseño visual y la arquitectura de la información**.

## ◆ Arquitectura de la Información (AI)

La **AI (Information Architecture)** organiza los contenidos para que el usuario **encuentre fácilmente la información** y navegue sin confusión.

### Beneficios de una buena arquitectura de la información

- **Mejora la usabilidad:** navegación clara y reducción de la carga cognitiva.
- **Aumenta la eficiencia:** acceso rápido a la información.
- **Reduce errores:** estructura lógica y comprensible.
- **Optimiza la experiencia de usuario (UX):** mejora la satisfacción y fidelización.

### Componentes clave

#### 1. Organización

- Definir estructuras jerárquicas, matriciales o secuenciales.
- Crear categorías lógicas (ejemplo: tipos de clientes en la web de Banco Santander).

#### 2. Navegación

- Diseñar sistemas que permitan moverse por la interfaz de forma intuitiva.
- Usar menús, barras, o **sistemas de “migas de pan” (breadcrumbs)** para mostrar la ruta actual.
- Planificar los **flujos de usuario**, es decir, los pasos que sigue el usuario para cumplir una tarea.

#### 3. Etiquetado

- Usar nombres y títulos claros y coherentes.
- Crear una **taxonomía** bien definida, como en el ejemplo de Fnac (clasificación de productos).

 El **ícono de búsqueda** es esencial en cualquier sistema bien diseñado: mejora la accesibilidad y permite al usuario llegar directamente a la información deseada.

## ◆ Wireframe

El **wireframe** es el **esquema estructural de la interfaz**, una representación visual inicial sin estilos ni colores.

Sirve para planificar la distribución de elementos y comprobar su coherencia antes del diseño final.

Principios fundamentales:

- **Simplicidad:** eliminar elementos innecesarios.
- **Funcionalidad:** cada componente debe tener un propósito claro.
- **Jerarquía visual:** destacar lo más relevante mediante tamaños, posiciones o contraste.

## ◆ Herramientas de trabajo

Para el desarrollo práctico de la asignatura se empleará:

- **Visual Studio Code** como entorno de programación principal.
  - HTML, CSS y JavaScript para la implementación del diseño y la estructura.
-